Learning Vector-Valued Functions

Massimiliano Pontil

Department of Computer Science
University College London, England

Joint work with Charles Micchelli (SUNY Albany, USA)

Outline

Let $\mathcal{Y}$ be a real Hilbert space with inner product $(\cdot,\cdot)$, $\mathcal{X}$ a set, and $\mathcal{H}$ a linear space of functions $f : \mathcal{X} \to \mathcal{Y}$. We assume that $\mathcal{H}$ is also a Hilbert space with inner product $\langle \cdot, \cdot \rangle$.

- Describe a model of functions $f : \mathcal{X} \to \mathcal{Y}$ which can be used for learning. In particular we focus on spaces of functions which admit a reproducing kernel.

- Extend regularization-based learning algorithms to this setting

- Show simulation studies.

Motivation: many applications!

Learning image transformations: image reconstruction, image de-noising, image morphing, etc. $\mathcal{Y}$ is the space of $n \times m$ matrices equipped with the Frobenius inner product.

Control theory: we want to learn a function over time, i.e. $f = (\ldots, f_{t-1}, f_t, f_{t+1}, \ldots)$ (discrete dynamical system)

Multiple labeled classification: $\mathcal{Y} = \{-1, 1\}^n$, $f = (f_1, \ldots, f_n)$, where f_j are binary classifiers.

Manifold Learning: we want to learn a $k-$dimensional manifold in $\mathcal{Y} \subset \mathbb{R}^n$ parametrized by $\mathcal{X} = [0, 1]^k$.

Learning output sequences: text-to-speech map, protein structure prediction. Both $\mathcal{X}$ and $\mathcal{Y}$ are spaces of finite sequences.

Example

Let $\mathcal{Y} = \mathbb{R}^n$, $(y, z) = \sum_{j=1}^n y_j z_j$, $y = (y_1, \ldots, y_n), z = (z_1, \ldots, z_n) \in \mathcal{Y}$.
We also write $f = (f_1, \ldots, f_n)$, where $f_j : \mathcal{X} \to \mathbb{R}$, $j = 1, \ldots, n$.

- **How to measure the "smoothness" of f?**

If each component belongs to the same RKHS of kernel k, we define

$$\|f\|^2 = \sum_{\ell=1}^n \|f_\ell\|_k^2$$

and learn the $f_\ell = \sum_{j=1}^m k(x, x_j) c_{j\ell}$ independently, e.g. by a LS–method.

Is this a good idea? maybe in some cases f is smooth if both its components *and* some combinations of them are smooth, e.g.

$$\|f\|^2 = \sum_{\ell=1}^n \|f_\ell\|_k^2 + \sum_{\ell=1}^{n-1} \|f_{\ell+1} - f_\ell\|_k^2$$

Let's try to make this observation more precise...

Reproducing kernel Hilbert spaces

Definition: *We say that $\mathcal{H}$ is a reproducing kernel Hilbert space when $\forall\ y \in \mathcal{Y}$ and $x \in \mathcal{X}$ the linear functional which maps $f \in \mathcal{H}$ to $(y, f(x))$ is continuous.*

A consequence of the above definition is that, for every $x, t \in \mathcal{X}$, there exists a bounded linear operator $K(x,t) : \mathcal{Y} \to \mathcal{Y}$ such that:

$$(y, f(x)) = \langle K(\cdot, x)y, f\rangle.$$

Typical function in $\mathcal{H}$: $f(\cdot) = \sum_{j=1}^{m} K(\cdot, x_j)c_j, \quad x_j \in \mathcal{X}, c_j \in \mathcal{Y}$

This property implies that

- $K(x,t) = K(t,x)^*$, and $K(x,x) \in \mathcal{L}_+(\mathcal{Y})$.
- For any $m \in \mathbb{N}$, $x_1, \ldots, x_m \in \mathcal{X}$, $c_1, \ldots, c_m \in \mathcal{Y}$:

$$\sum_{i,j=1}^{m} (c_i, K(x_i, x_j)c_j) \geq 0.$$

Note: one can show that if $K : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is a kernel then there exists a unique (up to an isometry) RKHS which admits K as the reproducing kernel.

Case $\mathcal{Y} = \mathbb{R}^n$

$K(x,t)$ is a $n \times n$ matrix with elements $K_{\ell q}(x,t), \ell, q \in \{1, \ldots, n\}$ and:

- $K_{\ell q}(x,t) = K(t,x)_{q\ell}$, and $K(x,x)$ p.d. matrix for every $x \in \mathcal{X}$.
- For any $m \in \mathbb{N}$, $x_1, \ldots, x_m \in \mathcal{X}$, $c_1, \ldots, c_m \in \mathcal{Y}$: $\sum_{j,\ell=1}^{m} (c_j, K(x_j, x_\ell) c_\ell) \geq 0$.

Some example of matrix-valued kernels

1. Diagonal Kernel: $D(x,t) := diag(k_1, \ldots, k_n)$, where k_j are scalar kernels. (easy)

2. $K(x,t) = AD(x,t)A^*$, where A is an $n \times n$ matrix.
   In fact: $K(x,t) = K(t,x)^*$ is immediate, and

$$\sum_{i,j=1}^{m} (c_i, K(x_i, x_j) c_j) = \sum_{i,j=1}^{m} (c_i, AD(x_i, x_j) A^* c_j) = \sum_{i,j=1}^{m} (A^* c_i, D(x_i, x_j) A^* c_j) \geq 0.$$

3. $K(x,t) = \sum_j A_j D_j(x,t) A_j^*$, where A_j are $n \times n$ matrices and D_j diagonal kernels.

My favourite example

Let $\mathcal{X} = \mathcal{Y}$ represents pixel images. Choose

$$K(x,t) = Se^{-\beta\|x-t\|^2} + Te^{-\delta\|x-t\|^2}$$

where $\beta \ll \delta$ and S is a nearly diagonal sparse p.d. matrix, T a sparse out-of-diagonal p.d. matrix.

If we write $S = AA^*$ and $T = BB^*$, the above argument shows that K is a kernel indeed.

- S models correlations between nearby components (pixels) of $f(x)$ and $f(t)$.
- T models correlations between far away components. Such correlations are strong only if two images are "close" (similar).

Important remark: This example shows that the matrix K cannot, in general, be reduced to a diagonal kernel, i.e. it is not possible to rotate the outputs so that K becomes a diagonal kernel!

My favourite example (cont.)

$$K_{\ell q}(x,t) = S_{\ell q} e^{-\beta\|x-t\|^2} + T_{\ell q} e^{-\delta\|x-t\|^2}$$

What is the inner product in $\mathcal{H}$?

$$\langle (f_1,\ldots,f_n),(g_1,\ldots,g_n)\rangle = \sum_{\ell,q=1}^{n} S_{\ell q}\langle f_\ell, g_q\rangle_\beta + T_{\ell q}\langle f_\ell, g_q\rangle_\delta$$

and this also tells us when f is smooth ($\|f\| = \sqrt{\langle f, f\rangle}$ small)

Low rank kernels

Consider the kernel

$$K(x,t) = AD(x,t)A^*, \ x,t \in \mathcal{X}$$

where A is a $n \times r$ matrices and D a $r \times r$ diagonal kernel. For r much smaller than n, this is a *low rank kernel*, e.g. if $r = 1$ $K(x,t) = AA^*$.

In many practical situations the components of f may be *linearly related*, that is, for every $x \in \mathcal{X}$ $f(x)$ lies on a linear subspace of $\mathcal{Y}$. In this case it is desirable to use a kernel which preserves this property.

Idea: use a *low rank kernel* modelled by the output examples:

$$K(x,t) = \sum_{j=1}^{m} y_j y_j^* k_j(x,t)$$

where $k_j, j = 1, \ldots, m$ are some scalar$-$valued kernels.

Regularization

Let $V : \mathcal{Y}^m \times \mathbb{R}_+ \to \mathbb{R}$ be a prescribed function and consider the problem of minimizing the functional

$$E(f) := V\left(f(x_1), \ldots, f(x_m), \|f\|^2\right), \quad f \in \mathcal{H}$$

Important case: given a set $\{(x_j, y_j) : j = 1, \ldots, m\} \subset \mathcal{X} \times \mathcal{Y}$ of examples, we choose

$$E(f) := \sum_{j=1}^{m} Q(y_j, f(x_j)) + \mu \|f\|^2$$

where $Q : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is a *loss function*.

Theorem: If for every fixed $y_1, \ldots, y_m \in \mathcal{Y}$, $V(y_1, \ldots, y_m, t)$ is strictly increasing in t and $f_0 \in \mathcal{H}$ is a local minimizer of $E(f)$, then

$$f_0(x) = \sum_{j=1}^{m} K(x, x_j) c_j$$

for some $c_1, \ldots, c_m \in \mathcal{Y}$.

Form of the solution

Theorem: If for every fixed $y_1, \ldots, y_m \in \mathcal{Y}$, $V(y_1, \ldots, y_m, t)$ is strictly increasing in t and $f_0 \in \mathcal{H}$ is a local minimizer of $E(f)$, then

$$f_0(x) = \sum_{j=1}^{m} K(x, x_j) c_j$$

for some $c_j \in \mathcal{Y}, j = 1, \ldots, m$.

Proof: (main idea) Reduce the minimization of $E(f)$ to the minimal norm interpolation problem:

$$\min\{\|f\| : f(x_j) = f_0(x_j), j = 1, \ldots, m, \ f \in \mathcal{H}\}$$

for which it is known that the solution has the above form (see paper). Then observe that the minimal norm interpolant is f_0 itself.

Square loss (regularization networks)

Square loss: $Q(y, f(x)) = \|y - f(x)\|^2$

In this case the coefficients $c_j \in \mathcal{Y}$ are the unique solution of the linear equations

$$\sum_{\ell=1}^{m} (K(x_j, x_\ell) + \mu\delta_{j\ell})c_\ell = y_j, \;\; j = 1, \ldots, m.$$

or

$$(G + \mu I)c = y$$

where G is a $m \times m$ block matrix, where each block is a $n \times n$ matrix, i.e. $G_{jk} = K(x_j, x_k), \;\; j, k = 1, \ldots, m$ (so G is a $mn \times mn$ scalar matrix),
and $c = (c_j : j = 1, \ldots, m), \;\; y = (y_j : j = 1, \ldots, m)$ are vectors in $\mathbb{R}^{mn}$.

SVM loss funtions

We consider the two cases

1. $Q(y, f(x)) := \sum_{\ell=1}^{n} \max(0, |y_\ell - f_\ell(x)| - \epsilon)$,

2. $Q(y, f(x)) := \max_{\ell=1}^{n} \max(0, |y_\ell - f_\ell(x)| - \epsilon)$.

The dual optimization problem can be easily worded out in such cases (see paper). The solution of this problem gives the coefficients c_j which appear in $f_0(x) = \sum_{j=m}^{m} K(x, x_j) c_j$.

Simulation study

Learn a target function $f_0(\cdot) = K(\cdot, x_1)c_1 + K(\cdot, x_2)c_2$
$c_1, c_2 \in \mathbb{R}^n, x \in [-1, 1]$

1. $K(x,t) = Se^{-6\|x-t\|^2} + Te^{-30\|x-t\|^2}$, $x, t \in [-1, 1]$, $S, T \in \mathcal{L}_+(\mathbb{R}^n)$ and compare the minimizers of $E(f)$ in $\mathcal{H}_G$ for

   Method 1 (to learn f_1): $G = K$.

   Method 2 (to learn f_2): $G = I\left(tr(S)e^{-6\|x-t\|^2} + tr(T)e^{-30\|x-t\|^2}\right)$

2. $K(x,t) = I\left(ae^{-6\|x-t\|^2} + be^{-30\|x-t\|^2}\right)$, $x, t \in [-1, 1]$, $a, b \geq 0, a + b = 1$ and compare the minimizers of $E(f)$ in $\mathcal{H}_G$ for

   Method 3 (to learn f_3): $G = K$

   Method 4 (to learn f_4): $G = \frac{a}{tr(S)}Se^{-6\|x-t\|^2} + \frac{b}{tr(T)}Te^{-30\|x-t\|^2}$.

Simulation study (cont.)

The training set always consisted of 20 points $\{(x_j, y_j = f_0(x_j) + \nu_j) : j = 1, \ldots, 20\} \subset [-1, 1] \times \mathbb{R}^n$ and was generated by sampling i.u.d. x_j and ν_j in $[-1, 1]$ and $[-a, a]^n$ respectively.

In each experiment we chose $n \in \{2, 3, 5, 10, 20\}$ and an output noise parameter $a \in \{0, 0.2, 0.5, 1\}$. For each pair (n, a), we computed the matrices $S = A^* A$ and $T = B^* B$ where the elements of matrices A and B were i.u.d. in the interval $[0, 1]$. The centers $x_1, x_2 \in [-1, 1]$ and the vector $c_1, c_2 \in [-1, 1]^n$ were also i.u.d.

Each simulation was repeated 100 times, that is for each pair (n, a), 100 test error measures were produced for method 1 and method 2.

Experiment 1

Table shows difference between MS-error between f_0 and f_2 and the MS-error of the difference between f_0 and f_1 with its standard deviation (so a positive value means that method 1 performs better than method 2)

$a \setminus n$	2	3	5	10	20
0	8.333e−7 (4.263e−7)	9.450e−6 (3.977e−6)	1.624e−5 (3.093e−6)	5.184e−3 (1.542e−3)	2.794e−2 (8.816e−3)
0.2	2.742e−5 (3.901e−5)	7.845e−4 (1.905e−4)	2.186e−3 (2.202e−4)	4.666e−2 (5.707e−3)	0.2896 (0.0736)
0.5	9.546e−5 (2.138e−5)	4.111e−4 (5.537e−5)	2.298e−3 (3.548e−4)	4.492e−2 (1.030e−2)	0.5017 (0.0973)
1	4.064e−4 (4.448e−4)	8.607e−4 (7.653e−5)	4.803e−3 (1.183e−3)	6.297e−2 (1.622e−2)	0.5913 (0.1540)

The non-diagonal kernel *always does better* than the diagonal kernel.

Experiment 2

Notation as before with method 3 (resp. 4) replacing method 1 (resp. 4).

$a \setminus n$	2	3	5	10	20
0	1.420e−4 (4.358e−5)	1.046e−5 (2.566e−6)	1.310e−7 (7.341e−7)	7.500e−6 (1.335e−6)	4.452e−5 (8.480e−6)
0.2	4.903e−5 (9.044e−6)	1.275e−4 (2.050e−5)	1.785e−4 (7.013e−6)	1.370e−4 (1.850e−5)	1.613e−4 (6.027e−6)
0.5	3.424e−4 (4.851e−5)	3.245e−4 (3.500e−5)	5.465e−4 (5.456e−5)	7.004e−4 (6.552e−5)	5.991e−4 (4.404e−5)
1	5.292e−4 (7.936e−5)	1.260e−4 (3.953e−5)	5.681e−4 (8.507e−5)	3.311e−4 (4.508e−5)	5.483e−4 (3.646e−5)

The non-diagonal kernel *always dose worse* than the diagonal kernel.

Experiment: Conclusions

Comparing Exp. 1 to Exp. 2 we are tempted to conclude that the non diagonal kernel is not recommended when the number of outputs is 2 or 3 (it does more damage in experiment 2 than benefit in experiment 1). On the other hand this type of kernel helps dramatically when the number for outputs is larger or equal to 5.

This nondiagonal kernel and its generalization may be important for applications where the number of outputs is large and complex relations among them are likely to occur.

More Info

C.A. Micchelli and M. Pontil, "On Learning Vector-Valued Functions", Research Note RN/03/08, Dept of Computer Science, UCL, July 2003.

available at `http://www.cs.ucl.ac.uk/staff/M.Pontil/~pontil`